FUTUREWEI
Technologies

Multicast for AI Data Centers

Emerging Standards and Open Problems

Mike McBride | Futurewei | ChiNOG | May 2025

AI-Related Work at the IETF: Overview

Six active areas spanning content preferences, agentic protocols, network infrastructure, and multicast transport

AIPREF WG -- Chartered

- Standard vocabulary for AI content-use preferences
- robots.txt + HTTP headers signaling for opt-out
- Chartered Jan 2025; scope revision after Sept 2025 WGLC
- Co-chairs: Krishnan, Nottingham

Agent Security -- Side Meeting

- Identity, auth, auditability for agents
- AUDIT WG proposed: interoperable auditing and accountability for AI agents across domains
- mailing list: agent2agent@ietf.org
- Reuse of OAuth, WIMSE, SCIM
- Open: dedicated AISEC WG vs. BCP document approach

Agentic AI Protocols -- Emerging

- Agent2agent and agent-to-tool communication standards
- Side meeting at IETF 123 Madrid; CATALIST BoF at IETF 125 Shenzhen
- Focus: Agent communication protocol, discovery, security, networking
- Security part closely related to OAuth/WIMSE/webbotauth WG

Multicast for AI (mcast4ai)

- P2MP transport for MoE, AllReduce, model distribution
- Two side meetings: Madrid 2025 + Shenzhen Mar 2026
- BIER as leading candidate; gaps in reliability + BitString scale
- Mailing list: mcast4ai@ietf.org

FANN -- RTGWG Adopted Draft

- Sub-ms congestion + failure signals to upstream nodes
- Bridges gap between hardware event detection and control plane
- Complementary to multicast feedback path (in-network ACK)
- draft-ietf-rtgwg-net-notif-ps-00

Agentic AI for Networks -- NMRG

- LLM/agent-driven network operations and management
- Network digital twin + agentic AI arch (NMRG)
- AI agent protocols for 6G / 3GPP coordination (I-Ds active)
- Research phase -- problem statements

These workstreams are currently independent but will need to coordinate as AI infrastructure matures

AI Training & Inference at Scale

Model Sizes

Hundreds of billions of parameters

GPT, DeepSeek (671B total parameters), Claude

GPU Clusters

Thousands of GPUs

Tightly coupled via high-speed fabric

Network Demands

400 Gbps+ links, ~µs latency

Near-zero tolerance for packet loss

Key Traffic Patterns in AI Data Centers:

- North-South: user traffic in/out of the DC — well-understood, well-served by unicast
- East-West (intra-DC): GPU-to-GPU communication during training and inference
- Collective operations (AllReduce, AlltoAll) dominate: same data to many receivers simultaneously

Why Unicast Falls Short

Today's approach: replicate at the source

Bandwidth Amplification

With N destinations, the source GPU sends N copies over the same uplink. At 400 Gbps with thousands of recipients, this exhausts source bandwidth and congests leaf-spine links.

Source CPU/GPU Overhead

Generating N copies shifts replication burden to the compute node — wasting precious GPU cycles that should be doing AI math, not network I/O.

Synchronization Barriers

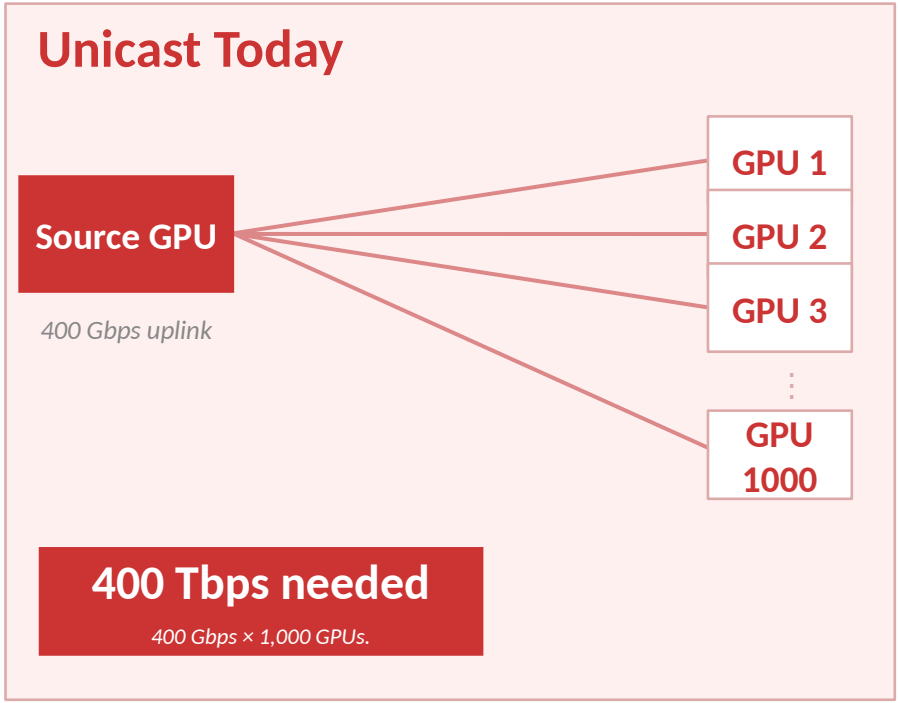
Collective operations like AllReduce can't proceed until the last recipient is ready. Unicast's non-simultaneous delivery creates stragglers — the slowest copy determines overall step time.

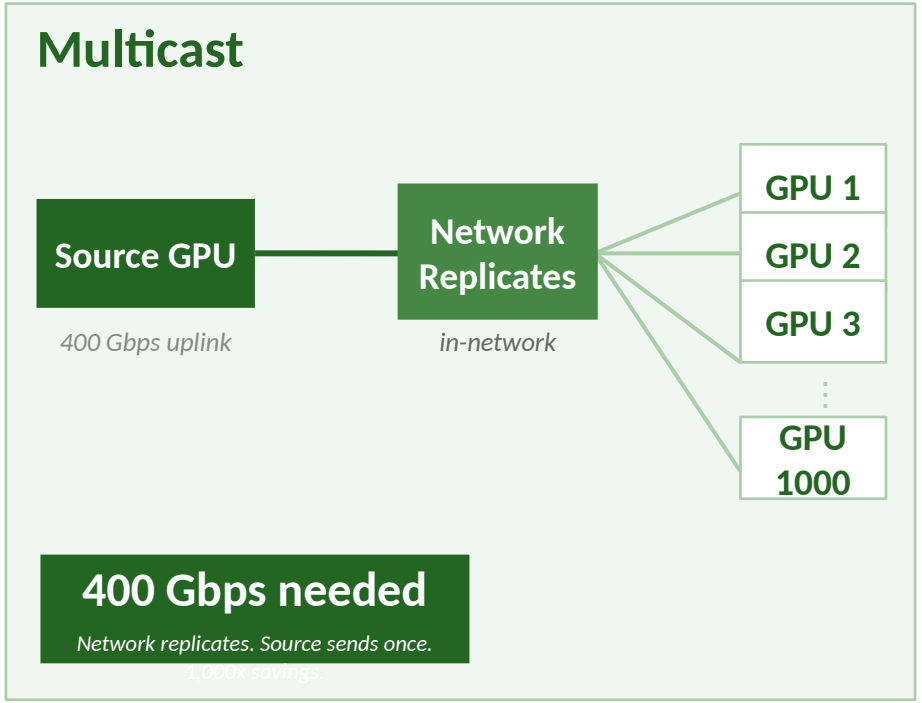
Scalability Wall

As GPU clusters grow from thousands to tens of thousands, unicast fan-out becomes untenable. The problem scales linearly with cluster size. The status quo is unsustainable.

Why Unicast Falls Short

Today's approach: replicate at the source





Mixture of Experts (MoE): The Key Use Case

What is MoE?

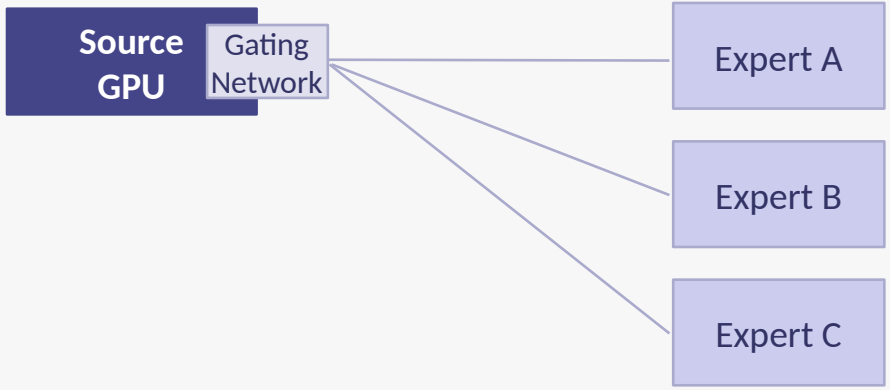
Instead of activating all model parameters for every token, MoE routes each token to a small subset of "expert" sub-networks. Only the selected experts are activated, dramatically reducing compute and cheaper to run.

Why this matters for the network: Each token dispatch is a P2MP operation — one source, multiple destinations, simultaneously. At 400 Gbps+ with thousands of GPUs, unicast copies of each token are a significant problem.

MoE Token Dispatch vs. IP Multicast

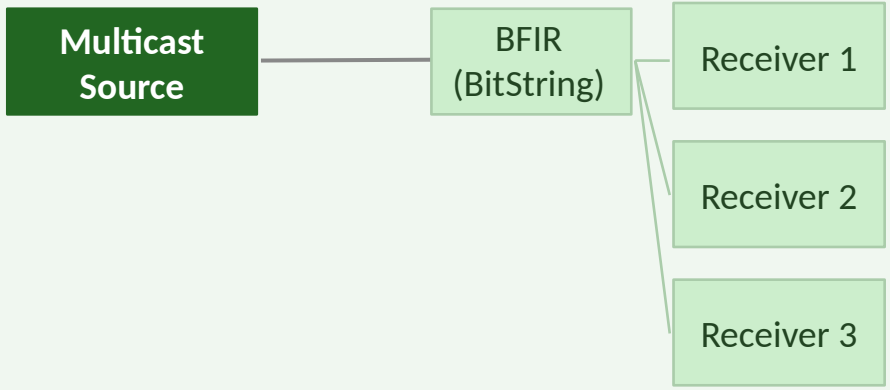
Both are fundamentally point-to-multipoint ie one source, same data, many receivers simultaneously.

MoE Token Dispatch



Source:	GPU holding the token
Data:	Same token to all selected experts
Receivers:	K experts — chosen dynamically by gating network
Timing:	Microsecond — no tree setup time available
Dynamics:	Group changes every token — extremely dynamic

IP Multicast (e.g. BIER)



Source:	Any host/router sending to a group address
Data:	Same packet to all group members
Receivers:	Group members — join via IGMP/MLD or BitString
Timing:	Tree setup (PIM) or in-packet (BIER)
Dynamics:	Stable (PIM) or per-packet dynamic (BIER)

Other P2MP Use Cases in AI Data Centers

AllReduce Broadcast Phase

Training

Distributed training uses AllReduce for gradient synchronization. Decomposed into Reduce + Broadcast phases — the Broadcast phase sends identical updated parameters to all participating GPUs. Classic P2MP pattern.

Model Distribution

Initialization

Before training begins, model weights (potentially terabytes) must be pushed from storage to all compute nodes. Unicast means N separate copies over the fabric. Multicast: one copy, replicated in-network.

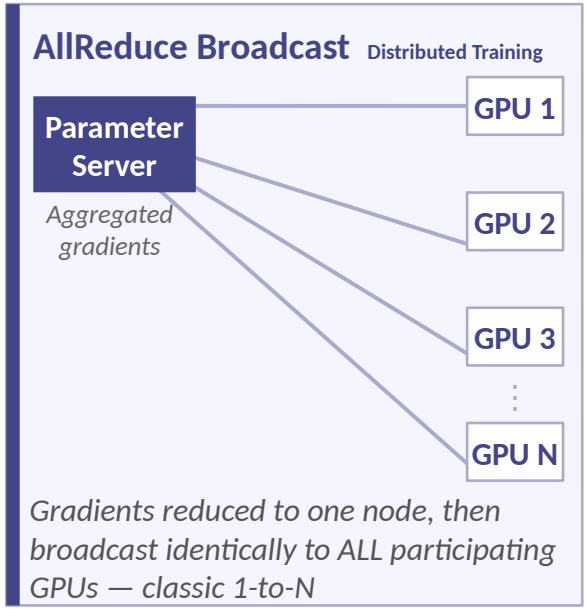
Multi-replica Checkpoint Storage

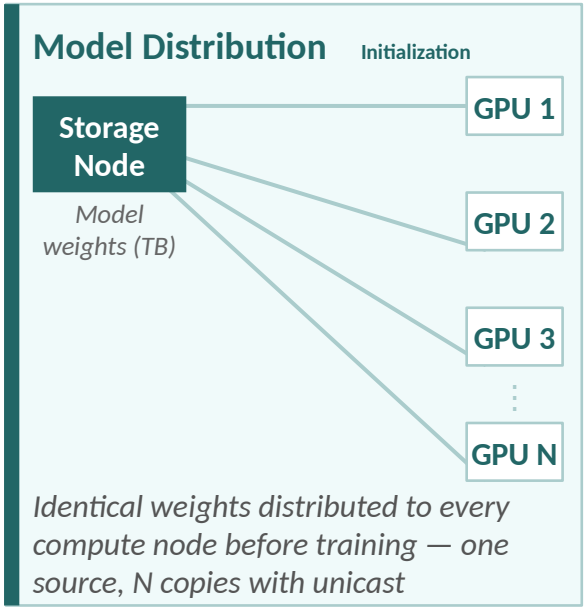
Reliability

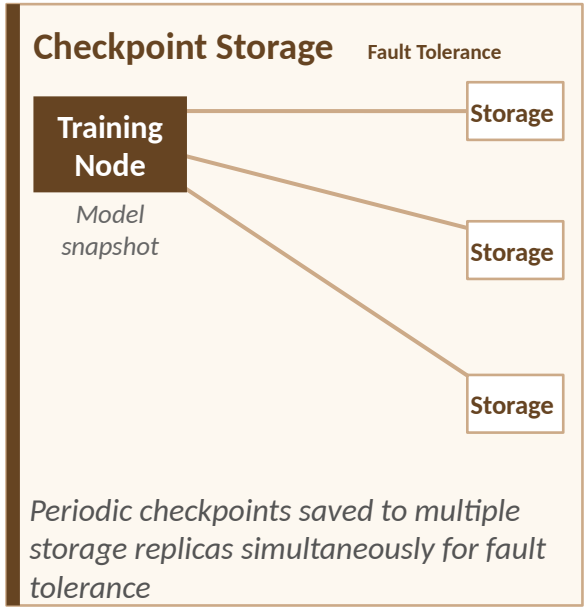
Periodic model checkpoints are saved to multiple storage nodes for fault tolerance. Same data, multiple destinations — another natural multicast use case.

Other AI P2MP Use Cases — Multicast Patterns

Each follows the same point-to-multipoint structure: one source, identical data, many receivers simultaneously.







All three share the same P2MP structure as MoE — multicast efficiency gains apply equally here

LLM Synchronization: Multicast for Inference Clouds

draft-liu-rtgwg-llmsync-multicast-00

Inference Cold Start: The Model Distribution Problem

Use Case

LLM inference is the dominant AI workload. **Cold-starting an inference instance requires downloading model weights (70 GB–1 TB+) before serving the first token**. Auto-scaling events, node failures, and traffic spikes trigger simultaneous cold-starts across many servers; all requesting identical data at once. With unicast, N separate copies exhaust source bandwidth and create severe latency precisely when capacity is needed most.

Inference & Distributed Inference — Beyond Training

Scope

Inference introduces distinct P2MP patterns: **model distribution to inference fleets, weight updates as new versions deploy and distributed inference across data centers** requiring synchronized model state across regions. These are more tractable than MoE token dispatch (stable groups, second-to-minute timescales) but the fan-out N is large and growing with fleet scale.

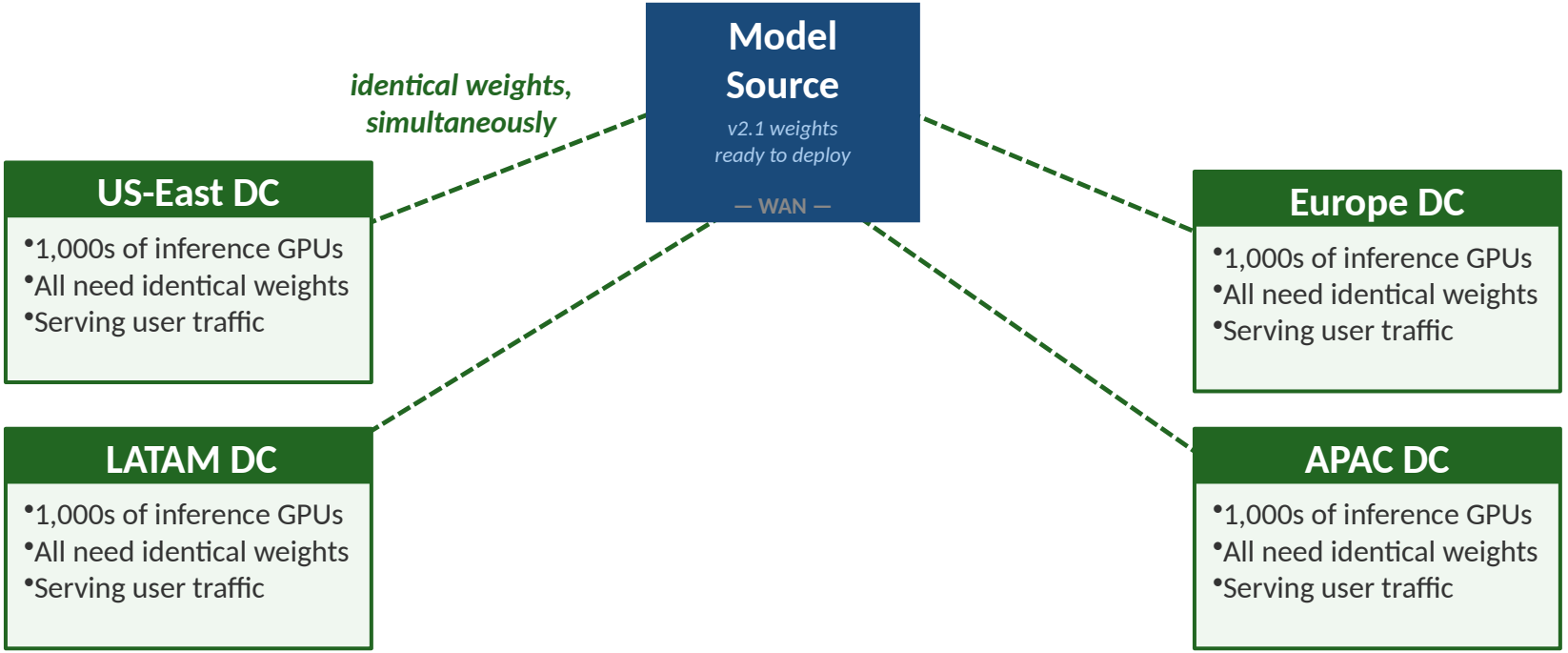
Why Multicast Fits Inference Synchronization

Solution Fit

Model distribution to inference fleets is a textbook P2MP use case: one source, identical data, many receivers — in-network replication eliminates N-copy waste. Inference group timescales (seconds–minutes) are within **PIM-SM and BIER** convergence bounds, unlike µs MoE dispatch. BIER is the best fit: stateless forwarding handles dynamic scaling events with no join latency. SR-P2MP is viable but heavier.

Inter-DC Inference Distribution: A P2MP Use Case

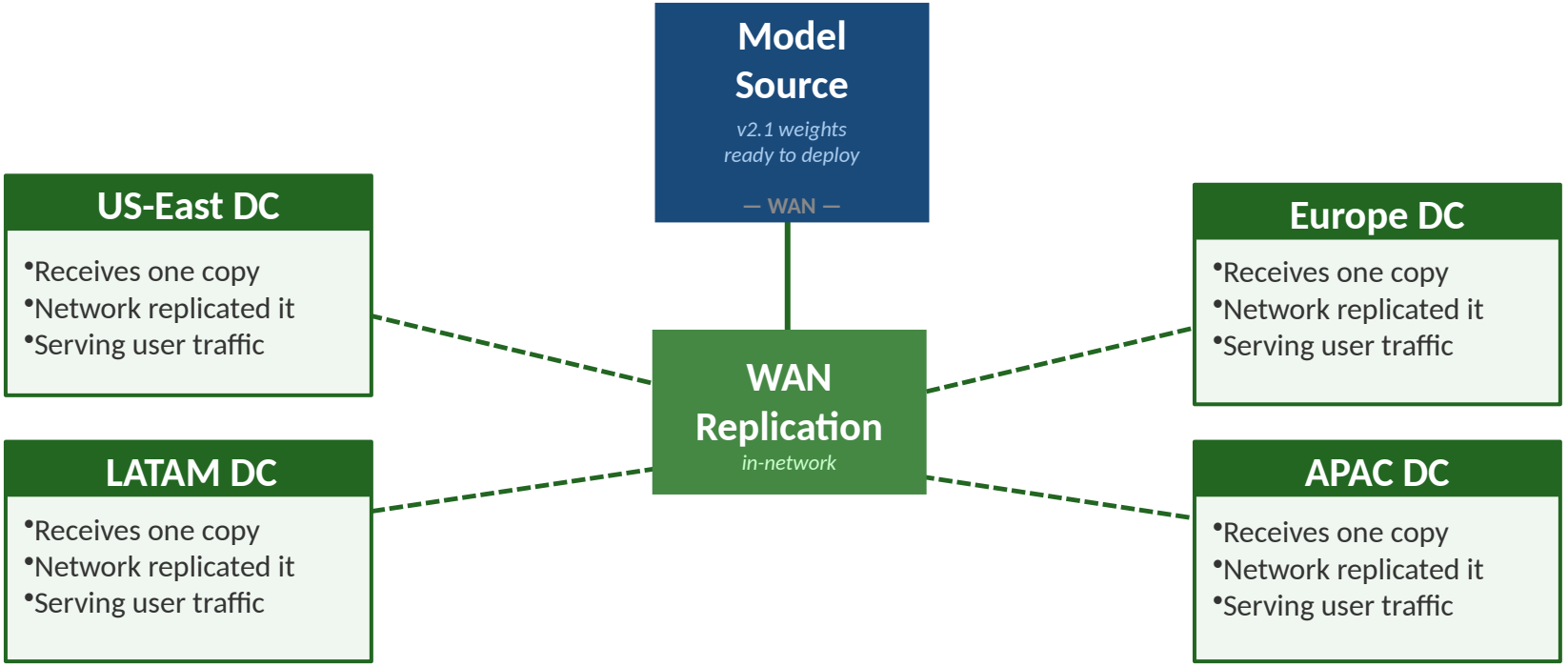
When a new model version deploys, identical weights must reach every regional inference fleet simultaneously — a textbook P2MP problem at WAN scale.



The P2MP argument at WAN scale: one source, identical weights, N regional fleets. Group membership is stable (minutes-to-hours). PIM, AMT and SR-P2MP are all viable. Unicast means N full copies over WAN links — expensive and slow.

Inter-DC with Multicast: Network Replicates

Source sends once. WAN replication point fans out. Each region gets an identical copy — without N separate WAN transfers.



Result: Source WAN egress drops from 4× to 1×. A 500 GB model = 500 GB over WAN instead of 2 TB.

Multicast Primer: Key Technologies

PIM-SM

Protocol Independent Multicast – Sparse Mode

✓ Widely deployed, well understood

✗ Tree-based state per group, slow convergence, control-plane heavy

Poor fit

mLDP

Multipoint LDP

✓ Works in MPLS environments

✗ Similar to PIM: per-tree state, slow adaptation to dynamic groups

Poor fit

SR-P2MP

Segment Routing P2MP

✓ Controller-driven, reuses SR forwarding

✗ Global recalculation needed on group change — too slow for µs dynamics

Moderate

BIER

Bit Index Explicit Replication (RFC 8279)

✓ Stateless — no per-flow state. BitString encodes receivers in header. Fast dynamic group changes

✗ BitString scales with domain size — efficiency degrades for sparse groups

Most promising

BIER: Bit Index Explicit Replication

RFC 8279 (2017) — Stateless multicast: receiver set encoded in the packet header, not stored in router state

Traditional Multicast (PIM)

- Receivers send IGMP/MLD joins to signal group membership
- Joins propagate through the network, building a multicast tree
- Each router stores per-group forwarding state (S,G) or (*,G)
- Source sends one packet — routers replicate along the tree
- Group changes require signaling and tree reconvergence

BIER — How It Works

- Each BFR in the BIER domain is assigned a BFR-ID for routing purposes. BFERs occupy a bit position in the BitString
- BFIR encodes the receiver set as a BitString in packet header
- Each router reads the BitString, replicates to neighbors with set bits, and clears forwarded bits
- No per-flow state at any router — forwarding is BitString-driven
- Group changes = change the BitString in the next packet. No signaling required.

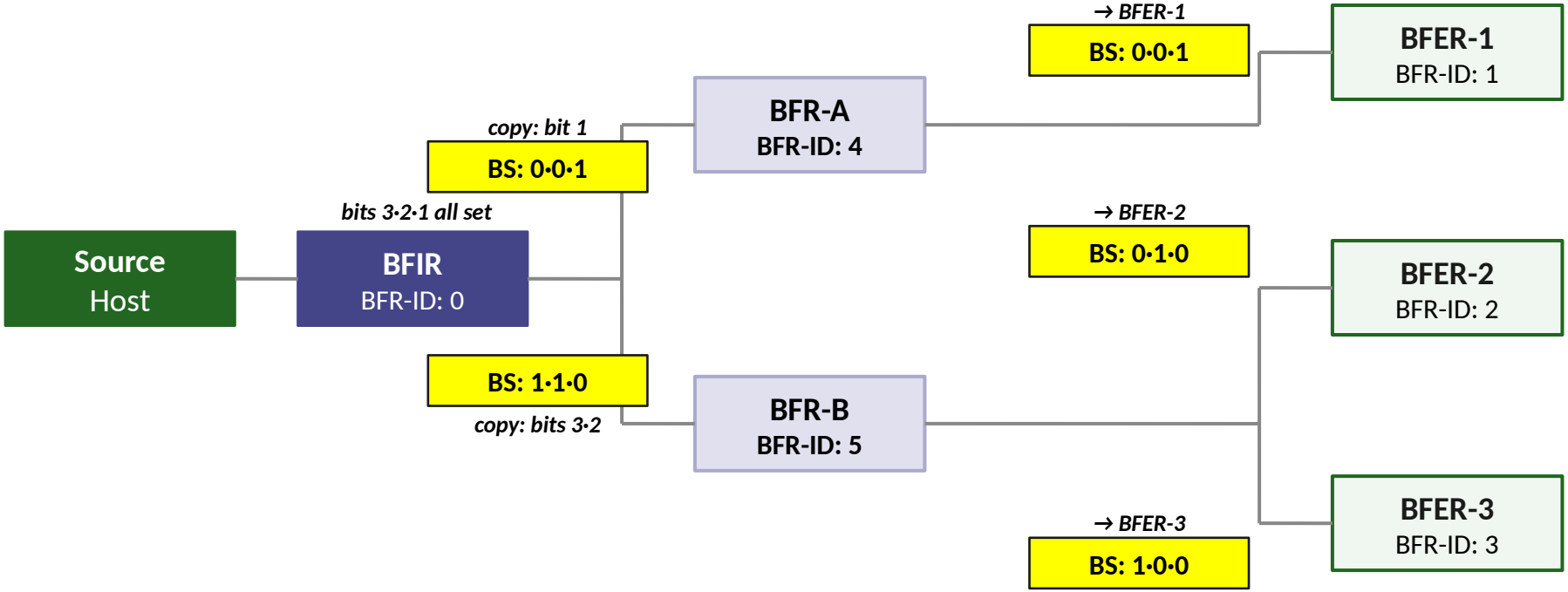
BFIR / BFER: Bit-Forwarding Ingress Router (BFIR): the first BIER-capable router, it attaches the BitString to the packet. Bit-Forwarding Egress Router (BFER): a router whose BFR-ID bit is set in the BitString — it is a destination for this packet.

BFR-ID & BitString: Every BFR in the BIER domain has a unique BFR-ID for routing purposes. Only BFERs occupy a bit position in the BitString — transit BFRs forward and replicate without consuming a bit. Current silicon typically supports 256-bit BitStrings; sub-domain partitioning extends coverage

No State — No Trees: Unlike PIM, BIER routers maintain no per-group or per-flow state. **The entire forwarding decision is made from the BitString in each packet header**. This makes BIER inherently stateless and immune to group-membership convergence delays. For MoE AI systems, the scheduler already knows the destination expert set, making expert selection conceptually similar to BitString construction

BIER: How the BitString Drives Forwarding

BFIR encodes BFER-1, BFER-2, and BFER-3 in the BitString and forwards into the BIER domain. BitString = 111 (00000000 00000000 00000000 00000111). Each BFR replicates and clears the bits it forwards



MoE / Scheduler logically becomes the Source and BFIR. Because the scheduler already knows the destination expert GPUs (BFERs), it directly encodes the BitString and injects the packet into the BIER domain.

Why BIER is the Focus — And Its Challenges

BIER: Stateless multicast via BitString encoding

How BIER Works

- Each node in the domain is assigned a BFR-ID for routing. BFERs, actual destinations, occupy a bit position in the BitString
- Sender encodes desired recipients as a BitString in the packet header
- Intermediate routers replicate packets based on BitString, no per-flow state maintained
- **Dynamic group changes = just change the BitString in the next packet. No tree setup, no signaling delay**

Gaps in AIDC Context

- BitString scales with domain size not with group size. Sparse MoE groups (9 of 256 experts) still carry a full-width BitString
- No native bidirectional support. ACKs and congestion feedback don't flow back through the multicast tree
- No native reliability mechanism. BIER is best-effort; AI workloads need near-zero packet loss
- In-network ACK aggregation not defined. Each receiver sending independent feedback causes feedback implosion

BIER is a good match for AI workloads, but it needs extensions to meet reliability and scalability requirements.

The Return Path: MP2P and Bidirectional Feedback

Why the Return Path Matters

AI training is zero-loss intolerant. **Receivers must ACK delivery so the sender can retransmit drops**. AllReduce barriers can't advance until all receivers confirm completion. A single straggler stalls the whole cluster. Without a return path, BIER is purely best-effort.

ACK Implosion: Why It's Hard

If 1,000 GPU nodes each send an independent ACK per multicast packet, **the source receives 1,000 ACKs per packet** — overwhelming at 400 Gbps rates. Loss events trigger simultaneous NACKs: a feedback storm far larger than the original event. BIER has no native return path; RFC 8279 defines only the forward BitString.

In-Network ACK Aggregation (draft-zzhang-bier-optimized)

Proposed: intermediate **BIER routers aggregate downstream ACKs** before forwarding upstream, reducing implosion from O(N) to O(tree depth). Each receiver ACKs its upstream router; feedback collapses toward the source through the tree.

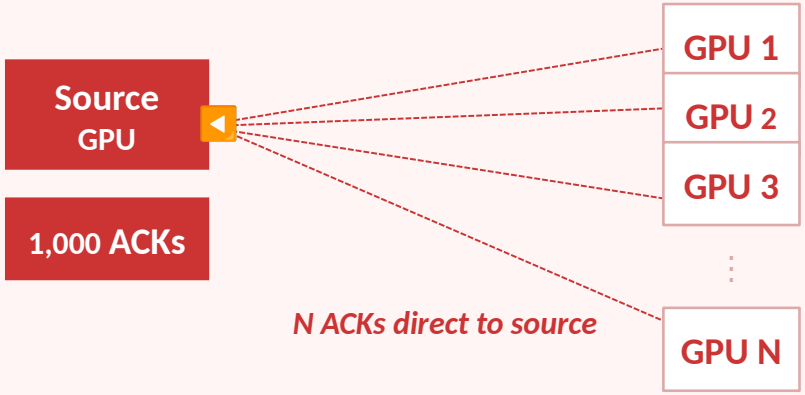
Other Approaches: BIDIR-PIM, FANN, Application-Layer ACK

BIDIR-PIM (RFC 5015) provides a native bidirectional shared tree, viable for stable training groups. FANN (draft-ietf-rtgwg-net-notif-ps) proposes sub-ms network-layer congestion notifications as a complementary return path**. Application-layer ACK via NCCL/oneCCL avoids new protocol work but fragments reliability across frameworks**.

MP2P: The ACK Implosion Problem and Solution

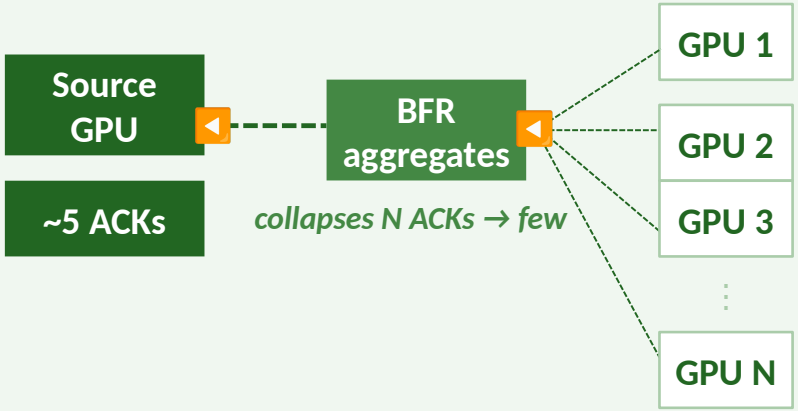
BIER delivers packets but has no native return path. AI training requires zero-loss confirmation. Here's why that matters.

Problem: ACK Implosion



N receivers → N independent ACKs → source overwhelmed. A loss event triggers simultaneous NACKs — a feedback storm larger than the original traffic.

Solution: In-Network ACK Aggregation



Each receiver ACKs its upstream BFR. ACKs collapse through the tree. Source receives O(tree depth) ACKs, not O(N). Reduces implosion from thousands to a handful.

Status: In-network ACK aggregation (draft-zzhang-bier-optimized) is proposed but not yet standardized. It is an open work item — one of the key gaps between BIER as currently defined and what AI data centers actually need.

Existing Technologies vs. AIDC Requirements

Source: draft-zhang-rtgwg-multicast-requirements-gaps-aidc

Technology	Interactivity	Reliability	Dynamics	Sparseness	Simplicity
PIM	Poor	Poor	Poor	Good	Poor
mLDP	Poor	Poor	Poor	Good	Poor
SR-P2MP	Poor	Moderate	Moderate	Good	Moderate
BIER	Poor	Moderate	Good	Poor	Good

No existing technology meets all AIDC requirements. A new architecture or extensions are needed.

- Interactivity: Bidirectional P2MP + MP2P feedback (ACKs, NACKs, congestion signals)
- Reliability: Lossless transmission, fast failure detection/recovery, minimized failure domain
- **Dynamics: Group membership changes at microsecond timescales with minimal overhead**
- **Sparseness: Efficient forwarding when only a small fraction of nodes are group members**
- Simplicity: Minimal control/data plane complexity for low latency and easy operation

IETF Activity: Where This Work Lives

Two Side Meetings So Far — Community Forming Around **mcast4ai**

IETF Side Meeting #1 | Madrid, 2025

Initial problem statement discussions. Established that AI workloads create new multicast requirements not met by existing protocols. Bootstrapped draft activity.

IETF Side Meeting #2 | Shenzhen, 2026

- Examined multicast requirements for MoE token dispatch, AllReduce broadcast, and model distribution
- Identified key needs: bidirectional feedback for reliability, dynamic group membership, scalability to thousands of GPUs
- **BIER recognized as promising foundation — gaps in reliability and BitString scalability acknowledged**
- Proposals discussed: in-network ACK aggregation, source-driven BIER optimization (eliminate receiver joins)
- Open question raised: **extend BIER vs. build new AI-specific multicast protocols**
- Coordination with **UEC (Ultra Ethernet Consortium)** discussed — need for quantitative performance data
- Agreed to continue discussion on mcast4ai mailing list. Send an email to **mcast4ai-join@ietf.org**

Key Multicast+AI IETF Drafts

Requirements & Gap Analysis of Multicast in AI Data Centers

Problem Statement

draft-zhang-rtgwg-multicast-requirements-gaps-aidc-01 | *Zhang, Cheng, Liu — March 2026*

Defines 5 key requirements for AIDC multicast (interactivity, reliability, dynamics, sparseness, simplicity) and scores PIM, mLDP, SR-P2MP, and BIER against them. Concludes no existing technology is sufficient — new work needed.

Optimized Use of BIER in AIML Data Centers

Solution Draft

draft-zzhang-bier-optimized-use-in-aidc-01 | *ZJ Zhang et al. — March 2026*

Proposes source-driven BIER optimization: sources know receivers (e.g., the gating network selects experts), so receiver joins via IGMP/MLD can be eliminated. Defines new IGMP/MLD 'Receiver Proxy Report' message so First Hop Routers can impose BIER encapsulation on behalf of source. Updates RFC 4604.

Multicast Usage in LLM MoE

Solution Draft

draft-zhang-rtgwg-llmmoe-multicast-01 | *Z. Zhang, Duan, Xu — October 2025*

Analyzes intra-node and inter-node multicast for token dispatching in MoE. Confirms BIER as best-fit for dynamic requirements. Notes that **NIC and collective communication library integration is needed for end-to-end deployment.** Ex: SLURM, the job scheduler, needs to manage multicast group lifecycle when allocating GPUs to training jobs.

Open Questions & Future Work

Q1

Extend BIER or build something new?

BIER is a good foundation but has fundamental gaps (interactivity, sparseness at scale). Is incremental extension sufficient, or does AIDC need a purpose-built protocol?

Q2

Reliability: where does it live?

In-network ACK aggregation is proposed but not yet standardized. Should reliability be handled at the network layer (new protocol work) or pushed to the application/collective communication library? Both approaches have tradeoffs.

Q3

BitString scalability at cluster scale

Current BIER BitString lengths top out at 256 bits. AI clusters are heading toward tens of thousands of GPUs. Hierarchical BIER, sub-domain partitioning, and 'unmasked BIER' are all being explored.

Q4

Quantitative performance data

The Shenzhen meeting explicitly called for benchmarks. Without measured data on latency improvement, bandwidth savings, and GPU utilization, it's hard to drive adoption. This is work the community needs.

Q5

SDO coordination: IETF + UEC

The Ultra Ethernet Consortium is working on AI transport from the hardware/link layer angle. Coordination between IETF IP-layer work and UEC fabric-level work is needed to avoid fragmentation.

Key Takeaways

1. **Large parts of AI workloads are fundamentally P2MP problems**

   MoE token dispatch, AllReduce broadcast, and model distribution are all P2MP operations. Unicast-based approaches are an inefficient workaround, not a solution.

2. **No existing protocol fully meets AIDC requirements**

   The IETF gap analysis shows that interactivity (bidirectional feedback) is a universal gap. BIER is the most promising starting point due to its stateless, dynamic nature. AI requires bidirectional group communication.

3. **BIER can be optimized for AI workloads**

   The source-driven approach (Receiver Proxy Report) eliminates join latency by exploiting a key property of MoE: the gating network already knows the receivers.

4. **The IETF community is actively engaged**

   Two side meetings, multiple drafts, a dedicated mailing list (mcast4ai). This is early-stage work that will need operator input, implementation experience, and performance data.

5. **Network operators should pay attention**

   AI infrastructure is your next major traffic engineering challenge. Understanding and contributing to this work now positions you ahead of when these deployments land.

Thank You.

Copyright © 2026 Futurewei Technologies, Inc.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Futurewei may change the information at any time without notice.

FUTUREWEI
Technologies